PCT/GB99/02517

RL

Europäisches Patentamt | European Patent Office | Office européen des brevets

EV

REC'D 23 AUG 1999
WIPO PCT

GB 99/2517

09/744393

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

98306106.0

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

DEN HAAG, DEN
THE HAGUE,
LA HAYE, LE 30/07/99

POTTE

EPA/EPO/OEB Form 1014 - 02.91

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 98306106.0

Anmeldetag:
Date of filing:
Date de dépôt: 31/07/98

Anmelder:
Applicant(s):
Demandeur(s):
BRITISH TELECOMMUNICATIONS public limited company
London EC1A 7AJ
UNITED KINGDOM

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
An index to a semi-structured database

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:
GB	30/07/98	

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
G06F17/30

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

AN INDEX TO A SEMI-STRUCTURED DATABASE

The present invention relates to a method and apparatus for generating an index to a semi-structured database containing a number of items, each item comprising a set of data stored in a semi-structured format, each set of data including a number of related entries.

Previously, there have been two major approaches to accessing data stored in electronic format. The first process is known as information retrieval and operates on a strict string search approach. Accordingly, if a user is to enter a query in the form of a keyword, using the information retrieval technique, the entire database will be searched for a string which matches the keyword. Obviously, such a system suffers from the drawback that it may well miss relevant entries should the form of the word in the database differ slightly to the form of the keyword. This problem can be overcome by using a stemming technique in which the keyword is truncated and a global word ending added. Again however this suffers from the drawback that numerous irrelevant records can then be located which include similar keywords.

In the second approach, known as knowledge representation, all the information from the database must be precoded using a special knowledge representation language to form a new database. This requires an operator to scan and analyze the data, placing relevant information in different knowledge representation fields. Once this has been completed, this allows users to access the information by entering queries in a knowledge representation language. This uses logic and theorem proving and is therefore not immediately accessible to users without specialised knowledge. In addition to this, knowledge representation approaches suffer from a drawback that the databases are initially hard to create and once created, even harder to change.

Both of the above mentioned techniques are anyway unsuitable for use with data stored in a semi-structured

format. A semi-structured database is a database in which some of the data within the database is stored in specific fields which denote the type of data whereas the remainder of the data will simply be stored under a general field, such as a free text field.

Databases of this form are generally created by either scanning in hardcopy records having predetermined formats, or having an operator enter data manually. However, because of the versatility of free text type fields, the data entered may vary in content and style. Whilst this reduces restrictions on the data that can be entered, making the database easier to create, it does mean that the different types of data stored cannot be determined by identifying the field in which the data is stored. Examples of cases were data is stored in such a semi-structured format include the Yellow Pages® directory, Exchange and Mart, Loot, and The British National Formulary.

Thus, for example, in the Yellow Pages® directory, the headings of various sections will be stored in a record that is designated as a heading field. Each individual advert (hereinafter referred to as an item) will include a name field and a free text field. A name entry is stored in the name field, whereas a free text entry, such as a description of the companies products or services, an address entry and a telephone number entry, will all be stored in the same free text field.

Accordingly, if information retrieval were applied to the Yellow Pages® directory, a search for a keyword would search through all the headings, company names and the free text. As the type of data is not accounted for, a heading may be located as a relevant result, when in fact the items associated with that heading are the results required. On the other hand, a knowledge representation technique of searching the database, would require that the database be translated into a separate knowledge representation database which could then be searched using knowledge

representation techniques. The original Yellow Pages® data would then be redundant, although if it were updated a new knowledge representation database would be required.

In accordance with the first aspect of the present invention, we provide a method of generating an index to a semi-structured database containing a number of items, each item comprising a set of data stored in a semi-structured format, each set of data including a number of related entries, the method comprising the steps of:

I) determining the presence of entries by comparing at least one set of data defining a respective item to each of a number of selection criteria, each selection criterion defining one or more predetermined characteristics of a respective entry; and,

II) generating a set of indices representing a concordance between the entries determined in step (I) and the respective items.

In accordance with a second aspect of the present invention, we provide apparatus for generating an index to a semi-structured database containing a number of items, each item comprising a set of data stored in a semi-structured format, each set of data including a number of related entries, the apparatus comprising:

a processor which determines the presence of entries by comparing at least one set of data defining a respective item to at least one of a number of selection criteria, each selection criterion defining one or more predetermined characteristics of a respective entry;

an index generator which generates a set of indices representing a concordance between the entries determined by the processor and the respective items; and

a data store which stores the set of indices.

The present invention provides apparatus and a method which generates an index to a semi-structured database. This involves defining a number of selection criteria which can be used to identify various entries in different fields of the semi-structured database. These selection criteria

are then compared to the items of the database so that specific types of entry can be determined within each item. An index is then generated which indicates the determined entry and the location of the respective item within the database. It is then possible to utilise these indices when searching to locate relevant items. Because the indices have a structured format, this allows a more versatile and efficient searching technique to be used.

Typically at least one set of data includes at least a free text field having a number of entries stored as a sequence of alphanumeric characters, wherein the predetermined characteristic of at least one of the entries is the format of a number of the alphanumeric characters. Thus for example, in the case of a Yellow Pages® directory, each item will generally include a telephone number within the free text field. The telephone number will be expressed as a sequence of digits which can only have a limited number of formats. Accordingly, by comparing the entire free text field to a set of predetermined formats, the telephone number can be located. Alternatively, the phone number may be provided in a specific phone number, field or be identified by virtue of being defined in a different font to the remaining text.

Typically each set of data includes a name entry, the predetermined characteristic of the name entry being that it is located in a name field. However, it will be realised that this is not essential. A name entry may not appear for example in the Exchange and Mart directory, or adverts in Loot or other similar advertising magazines. Even if a name entry is present, this need not be located in a known field and may be identified by some other means, for example, the font of the text, or the like.

Typically the method further comprises, for at least one set of data, defining any data not determined as an entry in step (I) as a free text entry. Thus, by selecting the remaining data from, for example, the free text field, this allows any remaining words in a free text entry to be

easily located. However, an alternative would simply be to compare the unmodified data to a list of keywords to locate any words that are believed to be relevant.

Typically the free text entry comprises at least one free text word defined by a sequence of alphanumeric characters. In this case, the method further comprises determining the presence of at least one selected free text word for a respective set of data by comparing the free text entry to at least one selection criterion defining one or more predetermined characteristics of a selected free text word; and, generating a set of indices representing a concordance between the selected free text words and the respective items.

Typically the predetermined characteristics of the at least one selected free text word comprise a predetermined list of words. This predetermined list of words may for example be a list of specific keywords. This allows selected words to be determined, such that words that would not be useful for searching, such as "the" for example, can be discarded. Alternatively however, the selection criterion could be such that only words which are verbs, nouns or adjectives are selected.

Typically the semi-structured database further comprises a number of heading fields each heading field preceding a number of related items and including at least one heading entry. In this case the presence of heading entries is determined by comparing each heading field to each of a number of selection criteria, each selection criterion defining one or more predetermined characteristics of a respective heading entry and generating a set of indices representing a concordance between the heading entries and the related items. This allows the headings of sections which may contain a number of related items to be searched. This is a more efficient searching technique as by identifying a relevant heading a large number of related items can easily be identified.

Typically at least one index in the set of indices indicates the location of an item within the semi-structured database having a respective entry determined in any of steps (I,IV,VI). It is also possible that at least one index in the set of indices indicates the location of each item within the semi-structured database having a respective entry determined in any of steps (I,IV,VI). Thus, each index may refer to one or more items depending on the format the indices are to take.

In accordance with a third aspect of the present invention, we provide method of identifying items contained in a semi-structured database having an index which has been generated according to any of the preceding claims, the method comprising the steps of:

A) generating a request for one or more items;

B) using the request and the set of indices to locate the one or more items within the semi-structured database; and

C) generating an output representative of the items. In accordance with a fourth aspect of the present invention, we provide apparatus for identifying items contained in a semi-structured database having an index generated by apparatus according to any of the preceding claims, the apparatus comprising:

an input for receiving a request for one of more items;

a processing device which uses the request and the set of indices to locate the one or more items within the semi-structured database; and

an output which generates a signal representative of the items.

Accordingly, we further provide a method and apparatus for identifying items contained within a semi-structured database. This operates by using a keyword determined from a user input request. This keyword is then compared to the entries listed in the index generated according to the first or second aspects of the invention allowing relevant

items to be located. The location of the item within the database is output, allowing the information within the item to be rapidly retrieved.

When identifying items contained in a semi-structured database, the step of locating the one or more items will usually involve the steps of analysing the request and determining therefrom at least one keyterm representative of the requested items, comparing each keyterm with each index, selecting the indices for items which have entries including the or each keyterm and using the indices to determine the location of each respective item in the semi-structured database.

An example of apparatus according to the present invention will now be described with reference to the accompanying drawings, in which:-

Figure 1 shows in schematic form apparatus for generating an index to a semi-structured database;

Figure 2a shows a typical item from the Yellow Pages® directory;

Figure 2b is a representation of the format of the data of the item of Figure 2a;

Figure 3 shows typical heading, see-reference and see-also-reference entries from the Yellow Pages® directory; and,

Figure 4 shows in schematic form apparatus for requesting an item from a semi-structured database.

Apparatus for generating an index to a semi-structured database will now be described with reference to Figure 1. The apparatus comprises a database store 1, which stores the data forming the semi-structured database to be indexed, and an index store 2 which stores the generated index. The index store 2 and the database store 1 are coupled to apparatus 3 for generating the index which will generally consist of a computer such as a SUN SPARC5-175 station, or the like. This includes a processor 4 coupled to a memory 5 which stores a number of predetermined

selection criteria. The processor 4 is also coupled to an index generator 6 via a bus 7.

Operation of the apparatus of Figure 1 will now be described. The semi-structured database stored in Figure 1 will generally include a number of items, each item being stored as a number of records. For example, in the case of the Yellow Pages® directory, each item 40 will generally comprise an individual advert, such as the advert shown in Figure 2A. This typically includes a name field 41 including a name entry 42 and a free text field 43 including a free text entry 44, an address entry 45 and a telephone number entry 46.

Each item in the database store 1 is stored as a number of records 51,52,53,54 with each record corresponding to a separate line in the item. Each record indicates in a first portion 51A,52A,53A,54A the item to which the record relates. A second portion 51B,52B,53B,54B indicates the type of field of the data. Thus, in the present example the second portion 52B,53B,54B of the latter three records will indicate that the data is provided in the free text field 43 and these will therefore be identical, whereas the second portion of the first record 51B indicates that the data is provided in the name field 41. The final portion 51C,52C,53C,54C of the records contain the actual data, such as the name entry 42, the free text entry 44, the address entry 45 and the telephone entry 46.

In use, the processor 4 will access the database store 1 to obtain the records 51,52,53,54 relating to a single item 40. The processor will then access the memory 5 to obtain one of a number of selection criteria. This selection criterion will be compared to the records 51,52,53,54 to locate a respective one of the individual entries within the item 40, which satisfies the respective selection criteria.

Once the entry corresponding to the respective selection criteria has been determined, the data relating

to that entry is extracted from the relevant record 51,52,53,54 and transferred to the index generator 6, along with an indication of the item with which the entry is associated. The index generator 6 then generates an index indicating the entry which was determined, and the item to which the entry relates. This is then transferred via the bus 7 to the index store 2. The processor 4 then accesses the memory 5 to obtain the next selection criterion.

Once each entry in the item has been indexed, the processor 4 accesses the database store 1 to obtain the next item in the database. The procedure is then repeated until all the items have been indexed.

It will also be realised that in the Yellow Pages® directory, the items 40 are arranged into sections of related items. As shown in Figure 3, each section 60 includes a heading entry 62 which is contained in a heading field 61. The heading entry indicates the nature of the related items and is provided with its own record.

Furthermore, there are also additional "see-reference" entries 63 and "see-also-references" entries 64 which may also be contained within the heading field 61 in respective records.

See-also-reference entries 64 are links to heading entries 62 of alternative sections 60 which may also include relevant items. See-reference entries 63 are again links to heading entries of alternative sections 60 that may include relevant items 40, however in contrast to the case of see-also-references entries, see-references entries are used when the section 60 including the see-reference entry does not in fact include any items. Accordingly, the heading entry, the see-also-reference entry and the see-reference entry are also transferred to the processor 4 for indexing.

In contrast to the indexing of items, each heading, see-reference and see-also-reference entry 62,63,64 does not include a specific item itself. Accordingly, once the processor 4 has located a heading entry, it must re-access

the data stored in the database store 1 to determine which of the items are located in the respective section. Details of these items are then transferred to the index generator 6 which will generate an index for the respective heading entry, the index including a list of the relevant items in the respective section. This list will also include a link to the heading entry of alternative sections, if there are see-also-references or see-references present.

The selection criteria themselves must be defined using an in depth knowledge of the database and the format in which the data is entered.

For example, in the case of the item 40 shown in Figure 2a, it is necessary to determine the presence of the name entry 42, the free text entry 44, the address entry 45, and the telephone number entry 46, as well as the heading entries. The procedure for achieving this will now be discussed separately for each entry.

Name Entry 42

This entry is readily identified as it is located in a specific name field and can therefore be identified by examining the record downloaded for the database store 1.

Telephone Number Entry 46

The location of telephone number entries 46 can be achieved by searching through the free text field 43 to locate a sequence of digits having a predetermined format. Thus, for example, in the item shown in Figure 2A, the telephone number entry 46 is "Colchester 822990". Accordingly, the selection criterion for locating the telephone number entry 46 will be designed to look for a town name followed by a six digit number.

Alternatively however the respective search criterion will also be needed to search for a four digit area code followed by a six or seven digit number. It is also necessary to take into account that there may be different

spacings between the digits in the phone numbers depending on the format used for entry of the telephone number. Accordingly, the search criterion which is used to locate telephone numbers preferably includes all possible telephone number formats, allowing any telephone number entry to be located.

Address Entry 45

Again, it is necessary to locate the address entry by comparing the free text entry to a number of likely formats for an address. Thus, in the example of Figure 2a, the address entry 45 could be located by searching for a 3 or 4 digit number followed by a word and then the term "street". Analysis of addresses shows that many do in fact contain terms such as road, street, avenue,... etc and accordingly, all these terms may be included in the selection criterion which is used for determining address entries.

In addition to comparing the free text filed for a term of this form and an address number, it is also possible to search for place names, such as Colchester. In this case such a search may not be successful as Colchester may have already been identified as part of the telephone number field 46. However, the aim is not to produce a single rule that will work for all items, but to produce a set of rules, each of which will be represented in the respective selection criterion, such that when the selection criterion is applied to the data, the relevant entry will be determined.

Free Text Entry 44

As far as the free text entry 44 is concerned, in the present example, this comprises the wording "suppliers of all top brand golf equipment". As this entry in itself is very difficult to locate, the processor 4 will determine the presence of a text entry 44 by firstly identifying and

then ignoring all the other entries in the free text field 43.

As the Yellow Pages® directory format is such that the free text field 43 will only ever include a text entry 44, an address entry 45 and a telephone entry 46, once these entries have been determined, the remaining alphanumeric characters left in the free text field 43 must comprise a free text entry 44.

In the case of the free text entry 44, this includes a number of words. Extraction of all these words would not be particularly useful for searching purposes. Accordingly, it is preferable to be more selective in choosing the words which are used to form an index.

One possible approach is to select a limited number of words from the text entry to form a list of keywords. An index may then be generated for each keyword. Thus, in the present example, the free text entry 44 is "suppliers of all top brand golf equipment". In this case words such as "of" and "all" are, in themselves, not very useful, and would therefore be discarded. In contrast the words "supplier" or "golf" form very good keywords.

However, the problem of selecting keywords is increased by the fact that there is no sentence structure in the free text entry, and that the upper and lower case distinctions which are used by many lexical analysis programs tend to be meaningless in these items. A solution to this is to have a predetermined list of keywords which are to be selected. This is however somewhat limiting, and it is therefore preferred to select words on the basis of certain properties.

In the present example this is achieved by deleting all words that are not nouns, verbs, or adjectives. These words can easily be identified using a system such as the "Brill-Tagger" which takes lines of words as input and tags the words with a part-of-speech tag indicating the nature of the words.

An index is then created for:

i) any single word tagged as a noun;

ii) any compound consisting of two or three consecutive words (i.e. where no intermediate word has been deleted); and

iii) noun compounds consisting of two or more words (these are indexed on the basis of any single word in the compound in combination with the 1st word).

The use of such compound keywords does have the limitation that many are too specific and may only relate to one item. This is overcome by deleting any compounds that are only associated with a single item.

As far as the remaining keywords and compounds are concerned, it is necessary to remember that there may be different varieties of the same word, such as golf, golfing, golfers. As a direct string comparison of golf and golfing will not produce a match, it is clearly preferable to modify the keyword or compound prior to forming the index.

Accordingly, the processor 4 accesses a lexicon such as "WordNet". This is used to convert any words located in the free text entry into their base form, such that golfing would be detected as golf. It is also possible to use stemmed forms of words, such as for example, "Lawnmow". This would then allow words such as lawnmowers, or lawnmowing to be detected.

A further alternative which needs to be considered when dealing with free text entries is the use of synonyms and hypernyms. These may be used to find words which are different but which have similar meanings. Thus for example a search for items relating to "teaspoons" may not locate very many records. However, if a search was carried out in the term "cutlery" then more records would be located. Accordingly, it is possible for the index to be created using more common synonyms or hypernyms of words to increase the number of relevant records that are located.

In some cases it is preferable to use a cyclic procedure to determine the free text entry. In this

operation successive amounts of text are deleted from the free text field until the number and form of the compounds and keywords which are determined are acceptable.

Heading Entry 62

As mentioned above, the heading entry 62 is identified by virtue of it being located in a heading field 61. Once identified however, it is necessary to select one or more keywords from the heading. This is performed in a manner similar to that used for the free text entry using the Brill-tagger, WordNet and a stemming routine. It is also necessary to ensure that any abbreviations in the headings are identified and modified into a keyword. This can be achieved by identifying the abbreviations in advance and ensuring the lexicon can identify the base form of the respective word from the abbreviation.

Once the indices have been defined, it is preferable to further define a set of ranking values indicating how relevant an item is to a particular index. This is achieved by determining the number of items that would be located using one specific index. In general, for the majority of indices, if a large number of items would be obtained, then each item has a relatively low ranking value indicating a relatively low relevance. In contrast, if only a small number of items are obtained for a particular index, these will have a high ranking value indicating that they are very relevant items.

The situation is further complicated by heading entries as each heading entry will refer to a number of related items, all of which are relevant. Accordingly, indexes for heading entries are given a higher ranking value than those for the text entries.

In the case of see-reference entries, the heading entry to which they refer is considered as though the original request referred directly to that heading entry. See-also-reference entries are however ignored as the

heading entries to which they refer are usually much more general than the heading entry under which the see-also reference occurs. Furthermore, there are often multiple heading see-also-references for any given heading entry.

It will however be realised that the calculation of ranking values is very much situation dependent, and the method employed will therefore vary for different semi-structured databases.

Apparatus for accessing the semi-structured database, using the generated indices, will now be described with reference to Figure 4, which shows in schematic form a system architecture for accessing items from a semi-structured database. The system, which will generally be formed from a computer device, includes a processor 100 coupled to an input/output device 101. The input/output device 101 may be any form such as a graphical user interface (GUI) and keyboard, or a microphone and speaker coupled to a speech recognition/synthesizer circuit.

The processor 100 is also coupled to a database accessing system 102. The database accessing system 102 includes a dialogue manager 103 which is coupled to the processor 100. The dialogue manager 103 is also coupled to a parser 104 and a query constructor 105. Both the parser 104 and the query constructor 105 are coupled to a world model 106 and to a backend 107. The backend is formed from the apparatus according to Figure 1 and therefore includes the semi-structured database store 1 and the index store 2.

In use, a request for information is supplied by a user using the input/output device 101. The request is transferred via the processor 100 to the dialogue manager 103 which operates to keep track of the current stage of the request processing, as well as controlling the operation of the parser 104 and the query constructor 105.

The request is passed to the parser 104 which modifies the request into a so-called slot-and-filler request, as will be explained in more detail below. This slot-and-

filler request is then transferred back to the query constructor 105.

The query constructor transforms the request into a database query using the world model 106. The query constructor then accesses the index store 2 in the backend 107 to obtain the location of relevant items within the database store 1. The relevant items are then located and transferred back to the processor 100 which will generate an output representative of the respective items.

Operation of each component of the system architecture 102 will now be described in more detail.

Dialogue Manager 103

The dialogue manager 103 is used to handle the requests input by a user via the input 101. The dialogue manager 103 will also monitor the request to determine if alternative operations have been requested. This may include operations such as quitting or re-starting, requesting help from the operating system, or correcting a request.

The dialogue manager 103 also monitors the results obtained to determine whether access of the database was successful and to monitor whether there are too many or too few located items.

Once the dialogue manager 103 has determined that a request has been made, the request is passed to the parser 104.

Parser 104

Requests are input into the system in the form of a standard sentence. Thus, there is not necessarily a standard structure to the request and it is therefore necessary to clarify the request by placing it in a form which the query constructor 105 can handle. This is done by modifying the request into a so-called slot-and-filler request, in which a series of predetermined fields filled

in using information derived from the request input by the user.

It will be realised that these fields will be entirely dependent on the nature of the database to be accessed. However, in the present example of the Yellow Pages® directory, suitable fields are:

1) Goods/Services;
2) Transactions; and,
3) Locations.

Thus, for example, if the user's input request is:

"I need to get my camera repaired"

the parser 104 will transform this into a slot-and-filler request, as shown below:

1) Goods/Services - Camera
2) Transaction - Repair
3) Location - <empty>

In the case in which phrases are included whose function is doubtful, for example propositional of phrases, these are generally put into the goods/services field.

Once the slot-and-filler request has been generated, this is passed to the query constructor 105.

Query Constructor 105

The query constructor 105 accesses the backend 107 and the world model 106 to determine a number of items which appear relevant to the slot-and-filler request. Thus, the query constructor 105 will access indexes containing the keywords entered in each field of the slot-and-filler request.

Thus, in the present example described above, the query constructor 105 would access any indices in the index store 2 that include the keywords "camera" and/or "repair". A list of any relevant items and their respective locations within the database store 1 is then returned to the query constructor 105 and passed onto the dialogue manager 103, which determines if there are sufficient or too many matches.

If there are insufficient matches, the query constructor 105 then operates to broaden the scope of the request. This is achieved using knowledge obtained from the world model 106, which generally includes a lexicon (for example "WordNet") including various synonyms, hypernyms, stemmed versions of words, and any other knowledge acquired by the user.

Thus, if the search results in too few matches, the query constructor 105 will access the world model 106 and determine a new keyword based on a synonym, hyponym or stemmed version of the original keyword. The search can then be repeated using the new keyword to obtain more results.

As an example, a query for teaspoons may not locate very many items. Accordingly, the query constructor 105 will access the world model 106 and determine that an equivalent word that could be used is cutlery. An enquiry for cutlery is then made with the backend 107, which will locate more items.

Similarly, if the request locates too many records, the query constructor 105 will narrow the search.

Once a suitable number of items have been located, the items will be transferred to the processor 100 and output via the input/output device 101. The actual output will also include an indication of the ranking values of the respective items, allowing the user to determine the relevance of the located item.

CLAIMS

1. A method of generating an index to a semi-structured database containing a number of items, each item comprising a set of data stored in a semi-structured format, each set of data including a number of related entries, the method comprising the steps of:

I) determining the presence of entries by comparing at least one set of data defining a respective item to each of a number of selection criteria, each selection criterion defining one or more predetermined characteristics of a respective entry; and,

II) generating a set of indices representing a concordance between the entries determined in step (I) and the respective items.

2. A method according to claim 1, at least one set of data including at least a free text field having a number of entries stored as a sequence of alphanumeric characters, wherein the predetermined characteristic of at least one of the entries is the format of a number of the alphanumeric characters.

3. A method according to any of the preceding claims, at least one set of data including a name entry, the predetermined characteristic of the name entry being that it is located in a name field.

4. A method according to any of the preceding claims, the method further comprising the step of:

III) for at least one set of data, defining any data not determined as an entry in step (I) as a free text entry.

5. A method according to claim 4, wherein the free text entry comprises at least one free text word defined by a sequence of alphanumeric characters, the method further comprising the steps of:

IV) determining the presence of at least one selected free text word for a respective set of data by comparing the free text entry to at least one selection criterion

defining one or more predetermined characteristics of a selected free text word; and,

V) generating a set of indices representing a concordance between the selected free text words determined in step (IV) and the respective items.

6. A method according to claim 5, wherein the predetermined characteristics of the at least one selected free text word comprise a predetermined list of words.

7. A method according to any of the preceding claims, wherein the semi-structured database further comprises a number of heading fields, each heading field preceding a number of related items and including at least one heading entry, wherein the method further comprises the step of:

VI) determining the presence of heading entries by comparing each heading field to each of a number of selection criteria, each selection criteria defining one or more predetermined characteristics of a respective heading entry; and,

VII) generating a set of indices representing a concordance between the heading entries determined in step (VI) and the related items.

8. A method according to any of the preceding claims, wherein at least one index in the set of indices indicates the location of an item within the semi-structured database having a respective entry determined in any of steps (I,IV,VI).

9. A method according to any of the preceding claims, wherein at least one index in the set of indices indicates the location of each item within the semi-structured database having a respective entry determined in any of steps (I,IV,VI).

10. A method of identifying items contained in a semi-structured database having an index which has been generated according to any of the preceding claims, the method comprising the steps of:

A) generating a request for one or more items;

B) using the request and the set of indices to locate the one or more items within the semi-structured database; and

C) generating an output representative of the items.

11. A method according to claim 10, when dependent on claim 8 or claim 9, wherein step (B) further comprises the steps of:

B1) analysing the request and determining therefrom at least one keyterm representative of the requested items;

B2) comparing each keyterm with each index;

B3) selecting the indices for items which have entries including the or each keyterm; and,

B4) using the indices to determine the location of each respective item in the semi-structured database.

12. Apparatus for generating an index to a semi-structured database containing a number of items, each item comprising a set of data stored in a semi-structured format, each set of data including a number of related entries, the apparatus comprising:

a processor which determines the presence of entries by comparing at least one set of data defining a respective item to at least one of a number of selection criteria, each selection criterion defining one or more predetermined characteristics of a respective entry;

an index generator which generates a set of indices representing a concordance between the entries determined by the processor and the respective items; and

a data store which stores the set of indices.

13. Apparatus according to claim 12, each set of data including at least a free text field having a number of entries stored as a sequence of alphanumeric characters, wherein the predetermined characteristic of at least one of the entries is the format of a number of the alphanumeric characters.

14. Apparatus according to claim 12 or claim 13, at least one set of data including a name entry, the predetermined

characteristic of the name entry being that it is located in a name field.

15. Apparatus according to any of claims 12 to 14, wherein for at least one set of data, the processor defines any data not determined previously as an entry as a free text entry.

16. Apparatus according to claim 15, wherein the free text entry comprises at least one free text word defined by a sequence of alphanumeric characters, wherein the processor determines the presence of at least one selected free text word for a respective set of data by comparing the free text entry to at least one selection criterion defining one or more predetermined characteristics of a selected free text word; and, wherein the index generator generates a set of indices representing a concordance between the selected free text words determined by the processor and the respective items.

17. Apparatus according to claim 16, wherein the predetermined characteristics of the at least one selected free text word comprise a predetermined list of words.

18. Apparatus according to any of claims 12 to 17, wherein the semi-structured database further comprises a number of heading fields, each heading field preceding a number of related items and including at least one heading entry, wherein the processor determines the presence of heading entries by comparing each heading field to each of a number of selection criteria, each selection criteria defining one or more predetermined characteristics of a respective heading entry and wherein the index generator generates a set of indices representing a concordance between the heading entries determined by the processor and the related items, the set of indices being stored in the store.

19. Apparatus according to any of claims 12 to 18, wherein at least one index in the set of indices indicates the location of an item within the semi-structured database having a respective entry determined by the processor.

20. Apparatus according to any of claims 12 to 19, wherein at least one index in the set of indices indicates the location of each item within the semi-structured database having a respective entry determined by the processor.

21. Apparatus for identifying items contained in a semi-structured database having an index generated by apparatus according to any of the preceding claims, the apparatus comprising:

an input for receiving a request for one of more items;

a processing device which uses the request and the set of indices to locate the one or more items within the semi-structured database; and

an output which generates a signal representative of the items.

22. Apparatus according to claim 21, when dependent on claim 19 or claim 20, wherein the processing device is adapted to analyze the request and determining therefrom at least one keyterm representative of the requested items;

compare each keyterm with each index;

select the indices for items which have entries including the or each keyterm; and,

use the indices to determine the location of each respective item in the semi-structured database.

ABSTRACT

AN INDEX TO A SEMI-STRUCTURED DATABASE

The present invention relates to a method of generating an index (2) to a semi-structured database (1). Semi-structured databases contain a number of items, each of which is stored as a set of semi-structured data including a number of related entries. The presence of these entries are determined by comparing the sets of data to a number of selection criteria, defining one or more predetermined characteristics of various entries. A set of indices is then generated representing a concordance between the determined entries and the respective items.

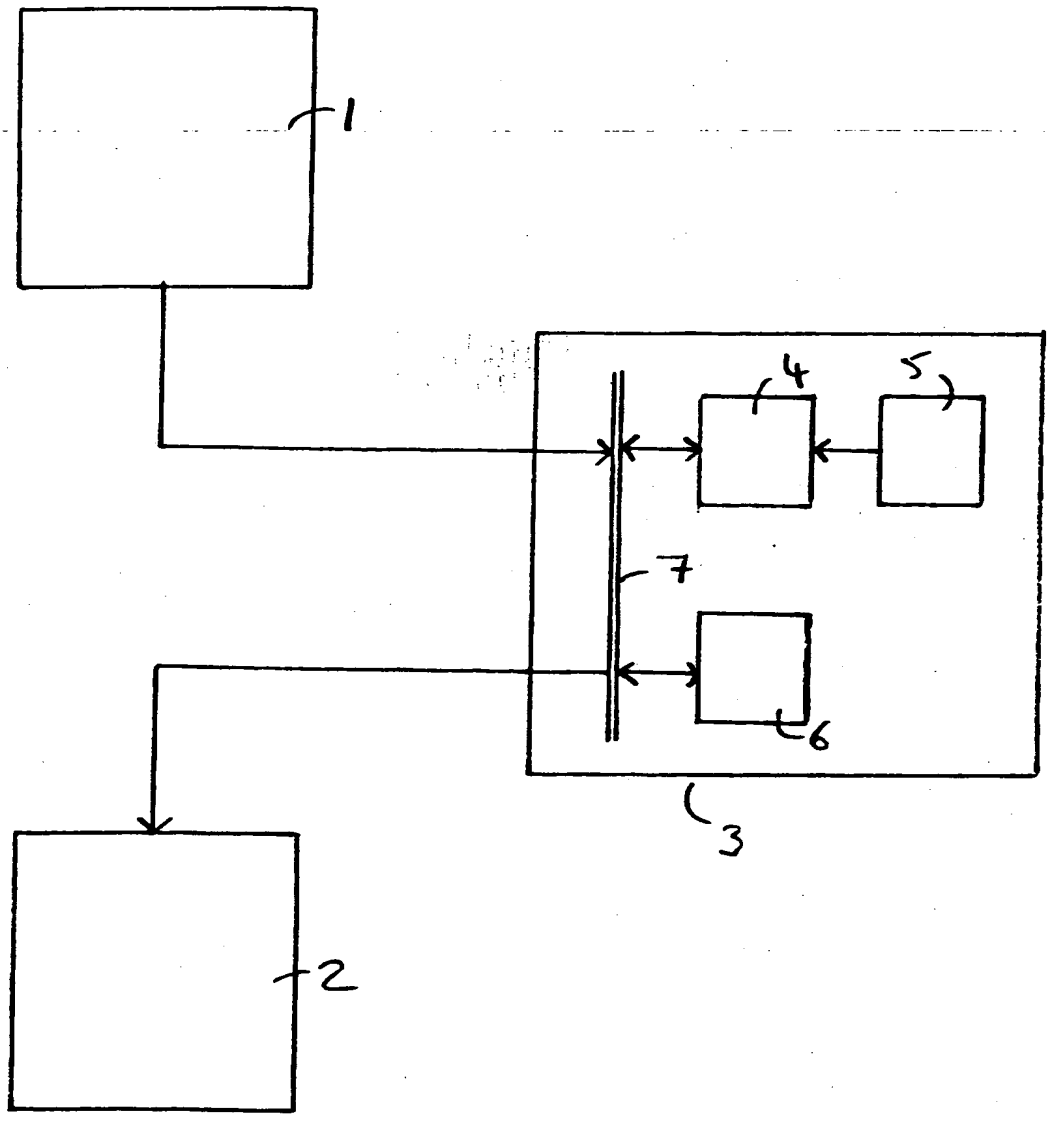

Figure 1

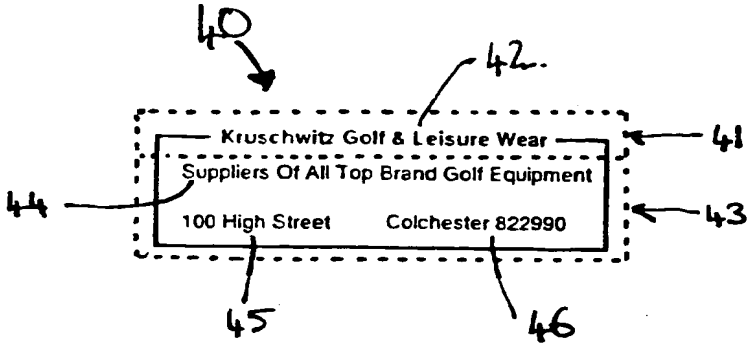

Figure 2a

51A	51B	51C
52A	52B	52C
53A	53B	53C
54A	54B	54C.

Figure 2b.

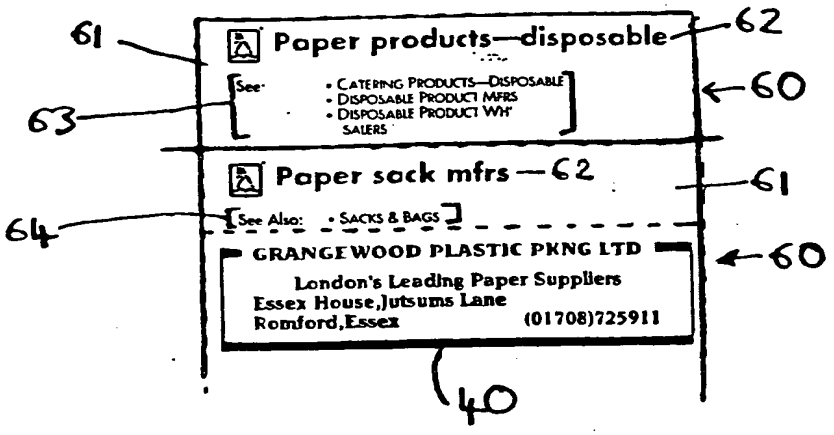
61
Paper products—disposable
62
See
• CATERING PRODUCTS—DISPOSABLE
• DISPOSABLE PRODUCT MFRS
• DISPOSABLE PRODUCT WH' SALERS
63
60
Paper sack mfrs — 62
61
64
See Also: • SACKS & BAGS
GRANGEWOOD PLASTIC PKNG LTD
60
London's Leading Paper Suppliers
Essex House, Jutsums Lane
Romford, Essex
(01708)725911
40

Figure 3.

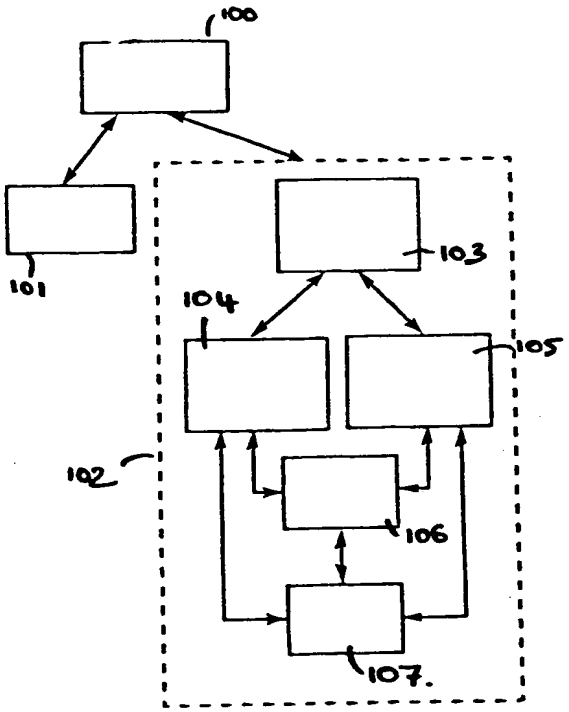

figure 4.